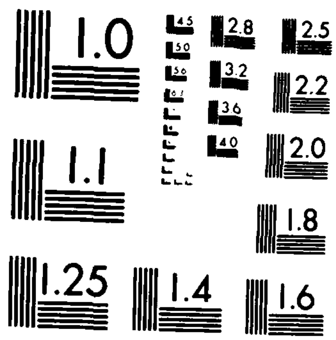
AD-A180 637 FINDING CAUSAL AND TEMPORAL RELATIONS IN EQUIPMENT FAILURE MESSAGES(U) NEW YORK UNIV NY COURANT INST OF MATHEMATICAL SCIENCES L JOSKOWICZ ET AL. FEB 87 1/1

UNCLASSIFIED PROTEUS-M-8 N00014-85-K-0163 F/G 12/5 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

DTIC FILE COPY

(2)

AD-A188 637

Finding Causal and Temporal Relations in Equipment Failure Messages

Leo Joskowicz, Ralph Grishman and Tomasz Ksiezyk

PROTEUS Project Memorandum #8
February 1987

PROTEUS PROJECT

Department of Computer Science
Courant Institute of Mathematical Sciences
New York University

2

Finding Causal and Temporal Relations in Equipment Failure Messages

Leo Joskowicz, Ralph Grishman and Tomasz Ksiezyk

PROTEUS Project Memorandum #8
February 1987

This report is based upon work supported by the
Defense Advanced Research Projects Agency under Contract
N00014-85-K-0163 from the Office of Naval Research.

Finding Causal and Temporal Relations in Equipment Failure Messages

Leo Joskowicz, Ralph Grishman and Tomasz Ksiezyk

Department of Computer Science
Courant Institute of Mathematical Sciences,
New York University
251 Mercer Street,
New York, NY 10012
joskowic@nyu-csd2.arpa

ABSTRACT

This paper describes the use of a simulation model to find causal and temporal relations between events in short equipment failure messages. A time independent mechanism to find the causal relations using both rule-based and simulation-based knowledge is presented. Using these causal relations, a causal chain is constructed. Various uses of the causal chains are discussed, in particular their use in finding temporal relations.

1. Introduction

The work presented here is part of the PROTEUS (PROtotype TExt Understanding System) project, whose objective is to understand short narrative messages about equipment installed in Navy ships. Casualty Reports (CASREPs) describe failures of this equipment, together with maintenance actions performed by the crew on board. A typical message consists of several sentences referring to equipment parts. For example, the message:

> Starting air regulating valve failed. Unable to consistently start number 1B gas turbine. Valve parts excessively corroded.

describes a failure of the air system due to the corrosion of some parts of a starting air regulating valve that controls the flow of air required to start the gas turbine. In order to build a ~~robust~~ system that understands this type of message, detailed knowledge of the domain of discourse is essential to the different phases of the text analysis (syntactic, semantic and discourse analysis). As one component of this knowledge, we have built a model of the equipment installed in the ship (initially for the Starting Air System). Previous papers have described this model-based approach [1, 2] and the use of the model in the interpretation of noun phrases [3]. In this paper we will concentrate on the issues related to discourse analysis, in particular finding causal and temporal relations between different facts.

Understanding failure reports is analogous to story understanding, as described by Schank in [4] and Wilensky in [5, 6]. Story understanding is viewed as an explanation driven process, where an event in a story is said to be understood only when a plausible explanation for that event with respect to other events in the story is found. In the stories about personal encounters in [5, 6], the characters of the story are assumed to have a particular goal and a plan to achieve it. Novel situations not included in the original plan may arise, and a means-ends analysis is performed to find a causal chain that explains the new situation. To build this causal chain, a

collection of rules capturing knowledge about beliefs and actions of people is used. This approach has several drawbacks for analyzing CASREPs. An exhaustive list of "failure plans" (analogous to the approach in [5, 6]) would amount to describing all possible faults which is an unreasonable task for large, complex systems. A possible solution is to use a few general "failure plans" and leave most of the explanation work to an inference system that is based on rules describing causal connections between events. The problem that arises then is to construct a coherent set of rules that captures the causal relations for the equipment in question. To solve this problem we propose to use a simulation model of the equipment that organizes knowledge about the equipment by capturing its causal, structural and functional relations. This simulation model is complemented by a set of hypothesis rules that are used to generate the hypotheses necessary to construct the causal chain.

In this paper we first present a short overview of the equipment model used in the analysis of CASREPs. We then provide a time independent mechanism to find causal relations using both rule-based and simulation-based knowledge. In the last section, time related issues are discussed, in particular the existence of two time modes in the messages and the use of the causal chain to find temporal relations.

2. The Equipment Model

Structural and functional knowledge about the equipment has been captured by an equipment model. In this model, the equipment is represented as a recursive transition network in which the nodes are equipment units (cooling system, lube oil pump, etc.) and the edges are conduits connecting these units. Conduits transmit medium, such as gas, liquids, mechanical movement or electric current. Every edge is associated with the media it transports. Figure 1 shows the top-level model network of the Starting Air Compressor (SAC). The SAC consists

of the air compressor, the motor system and the lube oil system. Each equipment unit is further described by another model network, establishing a hierarchy of model networks. Each equipment unit has a set of attributes describing the way the unit modifies input media, the different states in which the unit can be, the operational parameters related to it, etc. Using this information, the model is capable of carrying out simulations, i.e. propagating the consequences of a particular event throughout the network. We refer to the equipment model with its simulation procedures as the simulation model. This model and its use for noun phrase analysis is described in more detail in [3].

3. Causality

The problem of finding causal relations between events in a message can be stated in the following framework: let S be a set of *facts* such as states, events and observations. Our goal is to find all the subsets of S, S1, ... Sn, that are related by causal links. A causal link between two disjoint subsets of S, Si and Sj exists if the facts in Sj follow as a consequence from the facts in Si. In the CASREPs, S is the set of the facts either explicitly mentioned in the message or derived from them.

We distinguish between two methods of determining whether a set of facts is a consequence of another set: the rule method and the simulation method. In the rule method, causal links between facts are determined by using a collection of rules of the form:

IF *preconditions* and Si THEN Sj

That is, if *preconditions* are satisfied, Sj is a consequence of Si. Rules can also be used to deduce new facts; these facts can then be added to the initial set S and can be used to establish other causal links. The simulation method, on the other hand, presupposes the existence of a

model that is capable of performing simulations of facts. In this method, the simulation model is presented with the initial set of facts Si and the presumed consequences Sj. If the simulation of Si leads to Sj, then Si is said to be the cause of Sj. Links produced by the use of a single rule or a single simulation are called direct causal links and will be denoted by Si $\Rightarrow$ Sj, where Si is the antecedent and Sj the consequent.

There are two types of direct links: *deductive* links and *hypothesis* links. Deductive links are based on logically sound deductions connecting two sets of facts. They also provide a justification for facts that do not appear previously in the initial set of facts Si. For example, let the diesel motor, the motor system and the lube oil system be three equipment units in a rotary path, as shown in Figure 1. Assume that the diesel motor is rotating and that a gear in the motor system is sheared. Then we can deduce that no rotation is transmitted to the lube oil system, in particular to the lube oil pump. There is a deductive link between Si = {diesel motor rotates, gear sheared} and Sj = {lube oil pump does not rotate}. Hypothesis links on the other hand need not be logically sound, but are necessary to provide the required hypotheses to complete a causal chain in the absence of further evidence. For the previous example, assume that we know that the diesel motor is rotating and that the lube oil pump is not. Then we can hypothesize among several possibilities that a part in the motor system is sheared. A hypothesis link between Si = {diesel motor rotates, gear sheared} and Sj = {lube oil pump does not rotate} is then established.

We found it most appropriate to use the simulation method to find deductive links and to use the rule method to find hypothesis links. The simulation model is adequate to find deductive links since it has the capability of propagating the consequences of a set of facts throughout the model. Furthermore, all these consequences of this propagation are sound since

the simulation itself is sound. But the simulation model does not have the capability of producing hypothesis. It is therefore necessary to use hypothesis rules to generate new facts. It is the combination of these two methods that allows us to design an efficient algorithm for the construction of causal chains.

3.1. Simulation Queries

The simulation queries to the model are of the form MQ(E1, E2, C) where E1 and E2 are individual events that refer to a single equipment unit and C is the set of operational states of single equipment units, called the *context*. The interpretation of this query is: "Is there a causal relation between event E1 and event E2 in the context of the states C?"

The simulation is carried out as follows: first, the states of the different equipment units in C are set. Then, the possibility of E1 causing E2 is simulated, by looking for all paths between the equipment units mentioned in E1 and E2 and following the consequences of E1 along these paths. The answer to the query is then:

(1) *false* if there is no path between the equipment units of E1 and E2, or E2 is not a consequence of E1 given a context C. Otherwise,

(2) A list [<equipment-unit1, m_0>, <a_1, m_1> ...., <a_n, m_n>, <equipment-unit2, m_{n+1}>] where equipment-unit1 and equipment-unit2 are the parts referred to in E1 and E2 respectively and a_1, ..., a_n are the equipment units through which the event E1 propagates to cause E2; m_0, ..., m_{n+1} are the media through which the propagation takes place. There might be more than one such list, corresponding to the possible paths of propagation.

For example, given the actual (unedited) message:

While diesel was operating with SAC disengaged, the SAC lube oil alarm

sounded. Believe the coupling from diesel to SAC lube oil pump to be sheared. Pump will not turn when engine jacks over.

the PROTEUS semantic analyzer generates the following structures:

Message:
(event3, event4, event6)

EVENTS:
event(event3, (soundP @lo_alarm), (while state1))[1]
event(event4, (becomeP (shearedP @coupling)), NIL)
event(event6, (not (turnP @lo_pump)), (when state2))

OPERATIONAL STATES:
state(state1, (operateP @diesel), (with state3))
state(state3, (disengagedP @SAC), NIL)
state(state2, (jack_overP @diesel), NIL)[2]

QUERIES
1a. MQ(event3, event4, {state1, state3})
1b. MQ(event4, event3, {state1, state3})
2a. MQ(event3, event6, {state1, state2, state3})
2b. MQ(event6, event3, {state1, state2, state3})
3a. MQ(event4, event6, {state2})
3a. MQ(event6, event4, {state2})

ANSWERS
1a. false
1b. [<@coupling, rotation>, <@lo-pump, oil>, <@lo_alarm, oil>]
2a. false
2b. [<@lo-pump, oil>, <@lo_alarm, oil>]
3a. [<@coupling, rotation>, <@lo-pump, oil>]
3b. false

The scope of the context for each event and state is determined from the sentences' structure. Contexts are formed by collecting explicit references to states in events through the predicates with, while, etc. (a more refined approach would make use of time analysis to take also advantage of implicit relations). As we will see below, the answers to the queries can be used to con-

1. Notation: @part refers to equipment units.
2. "(jack_overP @diesel)" was treated as a synonym of "(operateP @diesel)".

struct a causal chain.

3.2. Rules

Hypothesis rules can be divided into two types: equipment dependent rules and equipment independent rules. Equipment independent rules are those rules that are true for any type of mechanism. Equipment dependent rules deal with specific types of equipment and capture their particular behavior. An example of an equipment dependent rule is the clutch rule:

[R10] Let [R, ... C, ..., A] be a rotary chain, where C is a clutch. Let R be an equipment unit that is rotating and A another unit that is not rotating. Then EITHER the clutch C is disengaged OR a part between R and C is sheared.

An example of an equipment independent rule is:

[G1] If a part X fails or is broken and X is part of Y then Y POSSIBLY fails.

Rules are indexed by the contents of their preconditions and stored in discrimination nets to allow their efficient retrieval. Testing some of the preconditions requires querying the equipment model. For example, in rule R10, to determine that specific equipment units e1, e2 and e3 mentioned in a message are all in the same rotary path, the equipment model is sent a query. These types of queries are called static queries as opposed to simulation queries, since they do not involve a simulation.

3.3. An algorithm to construct causal chains

By concatenating direct causal links such that the consequent of a link is the antecedent of the following link, we obtain a causal chain. The causal chain establishes a partial order among direct causal links, and provides an explanation of what happened. More specifically, given a set of facts S, a collection of sets A_1, ... A_n, B_1, ... B_n such that A_i, $B_i \subset S$, and direct causal

links between these sets, find a chain CH:

$$CH: \; A_1 \Rightarrow B_1, A_2 \Rightarrow B_2, \ldots. A_n \Rightarrow B_n$$

such that:

(1) $B_i \cap A_{i+1} \neq \varnothing$

(2) $A_i \cap B_j = \varnothing$, where $j > i$.

(3) CH is maximal, i.e. no other pair $A_k \Rightarrow B_k$ can be added without violating (1) or (2).

Condition (1) guarantees that there is a relation between the consequents in B_i and their use as antecedents in A_{i+1}. Condition (2) requires no loops to be present, i.e that an antecedent will not be later deduced (this excludes feedback phenomena, which produce infinite chains; as a first approach we ignore feedback phenomena). Condition (3) guarantees the smallest number of possible causal chains. The definition of a causal chain does not exclude the existence of several causal chains. There might be several explanations for a certain sequence of events, especially if some of the links of the chain are hypothesis links, or two unrelated sets of events happened simultaneously.

We can now present an algorithm to construct causal chains using simulation queries, static queries and rules. The input to the algorithm is a set of clauses in predicate notation, obtained from the Clause Semantics component (see [3]). There are two types of predicates: event predicates and operational state predicates. The first step of the algorithm classifies the equipment units of the message into sets, each set corresponding to the medium that the units process. Also, a list of abnormal events and states is established. Then, possible pairs of events, <E1, E2> are created together with their appropriate contexts C. Not all possible combinations of events are valid; for example, if the units mentioned in E1 and E2 are not related

in a path, then there is no need for a simulation query (the answer is always NIL). Once the simulation queries are prepared, they are sent to the simulation model. If it is possible to form from the answers to these queries a causal chain relating all events and states, the algorithm stops. If however there are events that are not connected to each other, there is a need for the hypothesis rules. The relevant rules are applied one by one, and the new facts are tested using simulation queries, to check for the consistency of the hypothesis. To avoid the rapid proliferation of hypothetical facts, we do not allow the application of hypothesis rules to hypothesized facts. The output of the algorithm is then a set of causal chains, where each link in a chain is tagged as either coming from a simulation query (and in which context) or from a hypothesis rule. The example presented above requires only simulation queries to build a complete causal chain:

From Query 1b: event4 $\Rightarrow$ event3 in context c1
From Query 2b: event6 $\Rightarrow$ event3 in context c2
From Query 3a: event4 $\Rightarrow$ event6 in context c3

Contexts: c1 = {state2}, c2 = {state1, state2, state3}, c3 = {state1, state3}

Causal Chain: event4 $\Rightarrow$ event6 $\Rightarrow$ event3

There are also cases in which an hypothesis is necessary. Consider for example the message:

> Unable to maintain lube oil pressure to SAC. Disengaged immediately after alarm sounded. Metal particles in oil sample and strainer.

the two facts "low lube oil pressure" and "metal particles in oil sample and strainer" do not have a direct causal link, so a simulation query would return NIL. The hypothesis of a faulty (broken, sheared or worn) part which drives the lube oil pump and from which the metal particles come is necessary to establish the causal link.

3.4. The uses of causal chains

What information can be extracted from causal chains? First, causal chains provide a possible explanation for what went wrong in the equipment; each step in the chain is an explanation of the sequence of events that lead to the malfunction. Causal chains also provide a means to identify the initial cause of the malfunction as well as all the subsequent malfunctions and faulty parts. The initial cause of malfunction is the first antecedent A_i of abnormal facts appearing in the chain; the ultimate consequence of the malfunction is the last consequent B_j of abnormal facts. In the example of section 3.1, the first abnormal situation is "coupling sheared", and the last consequence is "lube oil alarm sounds". This could be used to produce the Natural Language summary: "Sheared coupling between diesel and lube oil pump causes lube oil alarm to sound".

The causal chain also provides a validation for the coherence of the message since the causal chain is always consistent with the simulation model and establishes coherence relations between facts. The simulation model provides a tool to resolve ambiguities in messages: If messages contain an ambiguity, thus allowing two or more interpretations, all interpretations can be tested against the simulation model. The correct interpretation is the one which is consistent with the simulation model.

Another important use of the causal chain is in determining the temporal relations between events. Note that the causal chain provides a natural sequencing of events: if Si $\Rightarrow$ Sj then all the events in Si happened BEFORE all the events in Sj. In the next section we will discuss the use of causal chains in finding temporal relation.

4. Time Analysis

We distinguish between two temporal modes present in the CASREP messages: the time when things happened (Event time) and the time when things were discovered (Report time). Consider the previous message (section 3.3):

> Unable to maintain lube oil pressure to SAC. Disengaged immediately after alarm sounded. Metal particles in oil sample and strainer.

The first two sentences describe the events as they happened. First, the lube oil pressure could not be maintained. Then, the lube oil alarm sounded and the SAC was disengaged shortly afterwards. The third sentence describes the result of the investigation that took place after the alarm sounded and the SAC was disengaged. Nevertheless, the metal particles appeared *before* the alarm sounded, although the investigation revealed them only *after* the maintenance was performed. This corresponds to the following two temporal structures:

Event mode: fact4, then fact1, then fact3, then fact2.
Report mode: fact1, then fact3, then fact2, then fact4.

fact1: low lube oil pressure in SAC.
fact2: SAC disengaged.
fact3: Alarm sounds.
fact4: Metal particles in oil sample and strainer.

Both temporal structures are important since they serve different purposes: one describes actions from a maintenance point of view, the other from a causal point of view. Both temporal modes appear in most messages together, and it is necessary to distinguish between them.

Temporal relationships between facts in the message are either explicit or implicit. Explicit relationships are introduced by the use of the adverbs after, before, while, prior, etc. and by the tense of the verbs. Also, the the order of sentences within the message provides an ordering (but not always) of the events, since we can assume that the time does not move backwards in

narrative unless an explicit time marker is provided (see [7]). Using this information, we can extract the explicit time information for the Report mode. To complete this time information and to establish the temporal relations in the Event mode, causal and equipment dependent knowledge are required.

The operational characteristics of the equipment determine a definite sequencing between possible events. For example, the SAC must be working before starting the gas turbine; the lube oil alarm sounds after the lube oil pressure is measured as low; the SAC must be dismounted and disassembled before its drive shaft can be inspected. The sources of knowledge that establish this type of constraints on the sequencing of events are:

(1) Structural and functional information about the equipment.

(2) A state diagram describing the timing sequence and possible transitions between the states in which the equipment can be.

(3) Causal relationships between facts, and

(4) Time information derived from default values based on a small set of assumptions about how events within a narrative are connected temporally.

These sources of knowledge are useful for both the Event and the Report modes.

Temporal relationships can be captured by describing events and states that happen during intervals of time, in the way suggested by Allen in [8]. Each fact is associated with a time interval. Time intervals have a starting point and an ending point and can be related by 13 temporal relations. This set of relations together with a set of tables specifying how to compute their closure, constitute a qualitative calculus of intervals that is appropriate for our purposes.

The main source of information for the time structure in the Event mode is the causal chain. The causal chain establishes a temporal sequencing between events. Given a chain [$A_1 \Rightarrow B_1$, ... , $A_n \Rightarrow B_n$], all the facts in A_i occurred *before* the facts in B_i, all the facts in A_i occurred *before* the facts in A_j where $i < j$, and all the facts in B_i occurred *before* the facts in B_j where $i < j$. Although most of the temporal information is already present, there are some relations that are missing. In particular, those referring to the actions of the technician, not appearing in the causal chain. We present below a set of special temporal rules that can be used to establish such links.

As we associate the Event mode with the causal chain, we can associate the Report mode with the explicit syntactic information extracted from the message. The implicit relations can be reconstructed using special temporal rules.

4.1. Temporal rules

To complement the temporal links found by the causal chain and the explicit temporal information, we use special temporal rules. These rules describe timing sequences related to the actions of the technician and other implicit timing assumptions. The following are examples of such rules:

[T1] Maintenance events always happen after malfunction events.

[T2] Let A be a fact appearing in the text after a fact B. Then, unless otherwise specified, A also happened after B in the Report mode.

[T3] The detection of a malfunction (by inspection, alarm, etc.) happens always during or after the malfunction itself occurred.

Other temporal rules are already captured by the temporal calculus. For example, if I *after* J and J *after* K then I *after* K. These types of relations will appear when computing the transitive closure over the set of temporal relations.

5. Conclusion

We here described an explanation-based method of analyzing short equipment failure messages. This method is based on the use of both simulation-based and rule-based knowledge to construct causal chains relating events in the message. The use of a simulation model is indispensable for complex domains, such as the ship equipment domain. Message understanding requires a detailed knowledge of the possible interrelationships between events, and a simulation model provides the best means for capturing and organizing such knowledge.

Acknowledgement

This research was supported in part by the Defense Advanced Research Projects Agency under contract N00014-85-K-0163 from the Office of Naval Research.

References

[1] "Model-based Analysis of Messages about Equipment", by Ralph Grishman, Tomasz Ksiezyk and Ngo Thanh Nhan, PROTEUS Project Memo #2, NYU 1986.

[2] "Site Report: Research in Text Understanding" by Ralph Grishman and Lynette Hirshman, *Computational Linguistics*, Volume 12, Number 2, pp 141-145.

[3] "An Equipment Model and its Role in the Interpretation of Noun Phrases", by Tomasz Ksiezyk, Ralph Grishman and John Sterling (submitted for publication to IJCAI-1987).

[4] *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures* by R. Schank and R. Abelson, Lawrence Erlbaum Assoc., Hillsdale NJ 1977.

[5] "Why John married Mary: Understanding stories involving recurring goals" by R. Wilensky, *Cognitive Science 2,* 3 (July-Sept 1978, pp 235-66).

[6] "PAM and Micro PAM", by R. Wilensky in *Inside Computer Understanding*, R. Schank and C. Riesbeck eds., Lawrence Erlbaum Assoc., Hillsdale NJ 1981.

[7] "Representing Implicit and Explicit Time Relations in Narrative" by Lynette Hirshman and Guy Story, *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-81)* Vol 1, pp289-295, Vancouver, 1981.

[8] "Maintaining Knowledge about Temporal Intervals", by James Allen, *CACM*, 26 (11), 1983, pp. 832-843.

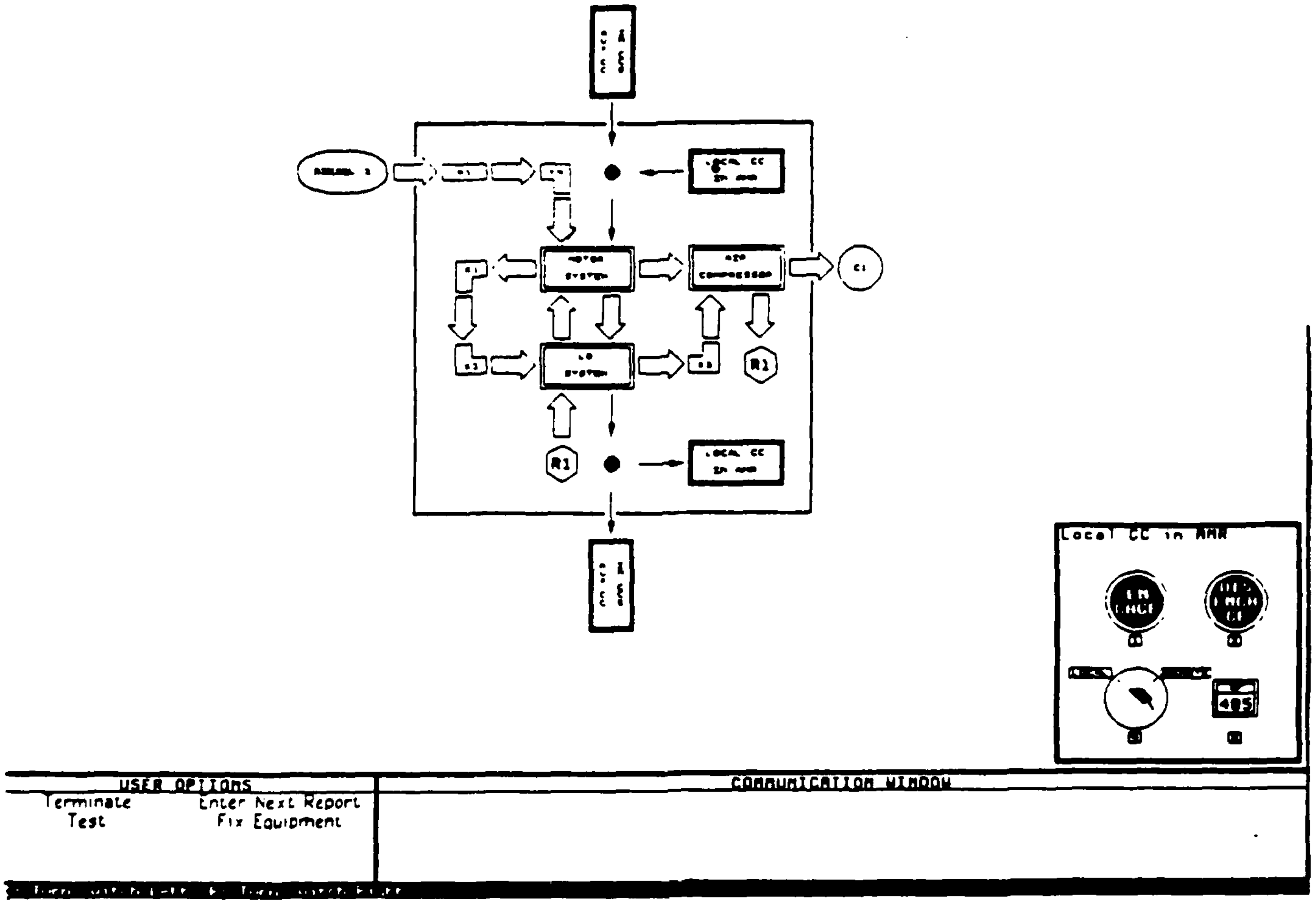


Figure 1 .Model network for SAC.

END

FILMED

MARCH, 1988

DTIC